Symmetrix Fibre Channel with QLogic QLA2200F-EMC Host Bus Adapters in the Linux X86 Environment

REV 3, JANUARY 2001

Copyright © 2000, 2001 EMC Corporation. All rights reserved.

No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of EMC Corporation.

The information contained in this document is subject to change without notice. EMC Corporation assumes no responsibility for any errors that may appear.

All computer software programs, including but not limited to microcode, described in this document are furnished under a license, and may be used or copied only in accordance with the terms of such license. EMC either owns or has the right to license the computer software programs described in this document. EMC Corporation retains all rights, title and interest in the computer software programs.

EMC Corporation makes no warranties, expressed or implied, by operation of law or otherwise, relating to this document, the products or the computer software programs described herein. EMC CORPORATION DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. In no event shall EMC Corporation be liable for (a) incidental, indirect, special, or consequential damages or (b) any damages whatsoever resulting from the loss of use, data or profits, arising out of this document, even if advised of the possibility of such damages.

Trademark Information

EMC^2 (the EMC logo), EMC, MOSAIC:2000, and Symmetrix are registered trademarks, and EMC Enterprise Storage, EMC Storage Logic, Celerra, Connectrix, CopyPoint, DataReach, Extended-Online, FarPoint, InfoMover, PowerPath, SDMS, SRDF, TimeFinder, The EMC Effect, and The Enterprise Storage Company are trademarks of EMC Corporation.

Adaptec is a trademark of Adaptec, Incorporated.

Red Hat is a registered trademark of Red Hat, Incorporated.

Linux is a registered trademark of Linus Torvalds.

All other trademarks used herein are the properties of their respective owners.

Introduction

This document describes the procedures for installing an EMC-approved QLogic host bus adapter (HBA) into a Linux® host and configuring the host for connection to EMC® Symmetrix® over Fibre Channel.

Check the EMC Open Systems Host Support Matrix for the latest information on approved HBAs and drivers.

Refer to the appropriate section for the installation procedures:

- Arbitrated Loop Configurations.....4
- Fabric Configurations 39

Arbitrated Loop Configurations

Useful Linux Utilities and Functions

These utilities and functions can be helpful in performing configuration operations:

Utility/Function	Description
fdisk	Command used to create and manipulate Linux partition tables
cfdisk	Curses-based disk partition table manipulator for Linux
fsck	Command used to check and repair a Linux file system
mkfs	Command used to create Linux file system on a device
mount	Command used to attach the file system on a device to the file tree
umount	Command used to detach a file system
shutdown	Command used to shut down the system gracefully
reboot	Command used to stop and restart the operating system
modprobe	Utility used to load or remove a set of modules that can be either a single module or a stack of dependent modules

Installing the HBA

Before the HBA is installed, the host must be configured with Linux. Install Linux from the Red Hat CD, following the procedure provided in the Red Hat installation guide. Partition the boot drive and select the packages and services necessary for the host.

To install the QLA2200F, follow the instructions included with your adapter. The adapter installs into a single slot, and has no user-configurable jumpers or switches.

Configuring Firmware Settings

After the HBA is installed, follow these steps to verify and configure HBA firmware settings.

Verifying the Correct Firmware Version

The firmware (HBA BIOS) version can be determined at boot time from the QLA2200F banner or from the **Fast!Util Options** menu.

- To determine the firmware version from the QLA2200F banner, boot the host and watch for the banner:

```
Qlogic Corporation
QLA2200 PCI Fibre Channel ROM BIOS Version 1.61
Copyright © Qlogic Corporation 1993-1999 All rights reserved
Press <ALT - Q> for Fast!UTIL
www.qlogic.com
```

Figure 1 QLA2200F Banner

  - If the ROM BIOS is v1.61, continue to *Setting the NVRAM Variables* on page 7.
  - If the ROM BIOS is not v1.61, upgrade the firmware as described under *Upgrading the HBA BIOS* on page 6; then go to *Setting the NVRAM Variables* on page 7.
- To determine the firmware version from the QLA2200 **Fast!Util Options** menu:
  a. Boot the host. As soon as the QLA2200F banner (see Figure 1) appears, press ALT-Q to enter the **Fast!Util Options** menu.
  b. Select **Configuration Settings** from the menu.
  c. Select **Host Adapter Settings** from the **Configuration Settings** menu.
  d. Under **Host Adapter Settings**, note the **BIOS Version**:
    - If the version is 1.61, continue to *Setting the NVRAM Variables*.
    - If the version is not 1.61, upgrade the firmware as described under *Upgrading the HBA BIOS* on page 6; then proceed to *Setting the NVRAM Variables*.

Upgrading the HBA BIOS

Follow these steps to upgrade the HBA BIOS:

1. Obtain the latest version of the BIOS from the QLogic Website:
   a. Access **http://www.qlogic.com.**
   b. Click **Driver Download.**
   c. Scroll to the bottom of the QLogic Drivers License Agreement and click **Use QLogic Drivers/Software**.
   d. Click **EMC Approved Drivers**.
   e. Click **EMC Approved QLA22xx Drivers**.
   f. Click **Link to ROM BIOS** in the **ROM BIOS v1.61** section.
   g. Save the file to a local directory as **22emc03bios161.exe**.
2. The file is a self-extracting ZIP file that must be extracted to a bootable diskette drive, as follows:
   a. Insert a diskette into a Microsoft Windows machine.
   b. Open any DOS window.
   c. At the DOS prompt, format the diskette:

      ```
      format /s a:
      ```

   d. At the DOS prompt, change (**cd**) to the directory in which you saved the zipped file, then extract the file to the diskette:

      ```
      22emc03bios161 a:
      ```

The QLA2200-EMC HBA must be installed in the host before proceeding.

---

3. Insert the BIOS upgrade installation diskette into the diskette drive.
4. Reboot the host.
5. After the host has rebooted, a DOS prompt appears. Type **flasutil /L /F** ENTER.

The HBA BIOS upgrade might take a few minutes.

---

6. After the upgrade is complete, remove the diskette and reboot the host. During boot-up, the QLA2200F banner (see Figure 1 on page -5) should display BIOS Version 1.61.

Setting the NVRAM Variables

Follow the steps below to set the NVRAM variables.

CAUTION

Any time the default settings for the adapter are chosen via Fast!Util, the NVRAM settings specific to EMC are overwritten. EMC recommends that you do not select the default adapter settings option from Fast!Util.

1. As the host boots, press ALT-Q when prompted `Press <Alt-Q> for Fast!Util`.
2. After the **Fast!Util** program loads, the display will depend on whether there are multiple QLogic HBAs installed:
   - If there is only one QLogic HBA, a **Fast!Util Options** menu will appear.
   - If there are multiple QLogic HBAs, a list of addresses occupied by those HBAs will appear. Using the arrow keys, select the desired HBA and press ENTER. The **Fast!Util Options** menu will then appear.
3. From the **Fast!Util Options** menu, select **Configuration Settings.**
4. On the **Configuration Settings** menu, select **Host Adapter Settings**.
5. Change the setting for **Frame Size** to **2048**.
6. Select **Host Adapter BIOS** and press ENTER to change the setting to **Disabled**.
7. Select **Adapter Hard Loop ID** and press ENTER to change the setting to **Disabled**.
8. Press ESC to return to the **Configuration Settings** menu.
9. Select **Selectable Boot Settings** from the **Configuration Settings** menu. Make sure that **Selectable Boot Device** is **Disabled**. If it is not, press ENTER to change the setting.
10. Press ESC to return to the **Configuration Settings** menu.
11. On the **Configuration Settings** menu, select **Advanced Adapter Settings**.
12. Increase the setting for **Execution Throttle** to **256**.

13. Change the setting for **LUNs per Target** to **128**. (The factory setting is **8**, but this must be changed to utilize all 128 LUNs available from the Symmetrix.)
14. Select **Enable LIP Full Login** and press ENTER to change the setting to **Yes**.
15. Select **Enable Target Reset** and press ENTER to change the setting to **Yes.**
16. Increase the setting for **Port Down Retry Count** to **32**.
17. Press ESC to return to the **Configuration Settings** menu.
18. On the **Configuration Settings** menu, select **Extended Firmware Settings**.
19. Change the setting for **Connection Options** to **0** (Loop only).
20. Press ESC to return to the **Fast!Util Options** menu.
21. When prompted to save the changes made to the current adapter, select **Save Changes**.
22. If you have more adapters to configure, choose **Select Host Adapter** and repeat steps 4 through 21 for each adapter.
23. Use the Fibre Disk utility to verify the disk media, as described under *Verifying the Disk Media* on page 9.
24. Press ESC to exit **Fast!Util**.
25. Reboot the host.

Verifying the Disk Media

To verify that the HBA can see the Symmetrix, use the Fibre Disk utility to scan and verify disk media.

1. Under the **Fast!UTIL Options** menu, select **FIBRE DISK UTILTY**.
2. You should see SYMMETRIX in a Fibre Channel device list of all possible 128 targets, as shown below. Page down if necessary to display all targets in the list.

```
   ID              VENDOR    PRODUCT     REV                     NODE NAME
<Hard loop ID>     EMC       SYMMETRIX   <micro code revision>   01000006048___
```

   - If SYMMETRIX appears in the list, go to step 3.
   - If SYMMETRIX does not appear in the list, verify the following, then start this procedure again at step 1:
     – that the correct Symmetrix port is connected to the host.
     – that the Symmetrix is on line.
     – that the cables are not damaged and are plugged in.
3. Use the arrow keys to select **SYMMETRIX** from the list of targets, then press ENTER.
4. A list of LUNs appears—select any LUN from the list.
5. A **Disk Utility Options** menu appears. Select **VERIFY DISK MEDIA**.
6. Select **CONTINUE WITH VERIFY**.
7. After media verification has completed:
   - If the message `Media Verification Complete` appears, press ESC repeatedly to exit **Fast!Util** and reboot the host, then continue with *Configuring the Host with the QLogic HBA Driver* on page 10.
   - If the media verification fails, verify that:
     – the Hard Loop IDs are unique between the HBA and the Symmetrix.
     – the cables and connectors are not damaged.

Configuring the Host with the QLogic HBA Driver

Using the QLogic adapter with the Linux operating system requires HBA driver software. The driver functions at a layer below the Linux SCSI driver to present Fibre Channel devices to the operating system as if they were standard SCSI devices.

Installation Methods

The QLogic driver can be installed onto a Linux host using one of two methods:

- Method One — Build the driver in the directory created for it.
- Method Two — Integrate the driver into the kernel.

If the driver is to be statically built in the kernel, the driver must be integrated into the kernel using Method Two.

There are advantages and disadvantages for each method of driver installation.

- Method One — Requires fewer file edits; however, Method One does restrict the use of the driver to modular only. The kernel must still be configured and compiled.
- Method Two — Requires additional edits to integrate the driver into the kernel and to incorporate the driver into **menuconfig**. Method Two is more flexible because it offers the ability to load the driver either as static or modular.

Both methods require editing **hosts.h** and **scsi.c**, and enabling **Probe All LUNs on each SCSI device** in **menuconfig**. Both methods require that the kernel be compiled.

Download the latest driver from the QLogic Website (refer to *Downloading the Driver* on page 12), then select one of the two methods to install the driver onto the Linux host:

- For a modular driver only, follow the procedure under *Installing the QLogic Driver — Method One* on page 13.
- To integrate the QLogic driver into the kernel so that it may be either modular or static, follow the procedure under *Installing the QLogic Driver — Method Two* on page 14.

Discussion of Default and EMC-Approved Drivers

EMC supports *only* the QLogic **qla2x00(smp)** driver, which can be downloaded from the QLogic Website.

EMC *does not* support the **qlogicfc** driver that is included in the Linux kernel (because the driver does not support persistent binding). Also, EMC *does not* support the alternate **qla2x00** driver that is included in the kernel in the Red Hat v7.0 distribution.

If the QLA2200F HBA is present in the system prior to a new installation of Linux, the Red Hat installation application automatically loads its version of the driver, which will be installed and inserted automatically when the host reboots. For this reason, EMC highly recommends installing the Linux operating system *before* installing the QLogic HBAs.

Removing the Default Driver

If the Qlogic QLA2200F is installed in the system prior to the Linux installation, the default driver *must* be removed from the kernel after rebooting, as follows:

1. Issue the appropriate command:

Linux Version	Enter:
6.2	`modprobe -r qlogicfc`
7.0	`modprobe -r qla2x00`

2. Edit the modules file to remove the unsupported driver:
   a. Issue the appropriate command:

Linux Version	Enter:
6.2	`vi /etc/conf.modules`
7.0	`vi /etc/modules.conf`

   b. Remove the appropriate entry:

Linux Version	Remove:
6.2	`alias scsi_hostadapter qlogicfc`
7.0	`alias scsi_hostadapter qla2x00`

Be aware that if the **qlogicfc** driver is alternately loaded and unloaded with the EMC-Approved QLogic **qla2x00(smp)** driver, there may be problems seeing Symmetrix devices. The two drivers use different firmware revisions; therefore, alternately loading both drivers might have an adverse effect on the HBA.

Downloading the Driver

Use the following procedure to download the QLogic driver from the QLogic Website:

1. Create a directory to which you will save the driver:

   ```
   mkdir /home/qlogic
   ```

2. Use a Web browser to access **http://www/qlogic.com.**
3. On the QLogic home page, click **Driver Download.**
4. Scroll to the bottom of the QLogic Drivers License Agreement and click **Use QLogic Drivers/Software**.
5. Click **EMC Approved Drivers**.
6. Click **EMC Approved QLA22xx Drivers**.
7. Click **Link to Source files** in the **Linux driver v2.19.15** section.
8. Save the driver to a local directory you created.

Select one of the two methods to install the driver onto the Linux host:

- For a modular driver only, follow the procedure under *Installing the QLogic Driver — Method One* on page 13.
- To integrate the QLogic driver into the kernel so that it may be either modular or static, follow the procedure under *Installing the QLogic Driver — Method Two* on page 14.

Installing the QLogic Driver — Method One

Follow these steps to configure the driver as modular only:

1. Change to the directory to which the tar archive was saved.

   ```
   cd /home/qlogic
   ```

2. Uncompress and extract the source files from the tar archive:

   ```
   tar zxvf qla2x00src-v2.19.15.tgz
   ```

   The files are:

   - **Makefile**
   - **ql2100_fw.h**
   - **ql2200_fw.h**
   - **qla2100.h**
   - **qla2100.c**
   - **release.txt**
   - **emc.txt**

3. Create the object file and install it into the appropriate directory of **/lib/modules/$1/scsi** (where **$1** is the kernel version).

   To build the driver, enter the appropriate command:

System:	Enter:
UP	`make install`
SMP	`make install SMP=1`

4. To load the QLogic driver on the system, enter the appropriate command:

System:	Enter:
UP	`modprobe qla2x00`
SMP	`modprobe qla2x00smp`

5. In order to load the driver automatically at boot time, **/etc/conf.modules** must be edited:

   ```
   vi /etc/conf.modules
   ```

   Add the appropriate entry:

System:	Enter:
UP	`alias scsi_hostadapter qla2x00`
SMP	`alias scsi_hostadapter qla2x00smp`

6. Run the following command to update your **/lib/modules/$1/modules.dep** file (where **$1** is the kernel version) with the information you added to **/etc/conf.modules**:

```
depmod -a
```

7. Proceed to *Required Kernel Modifications* on page 22 to modify and compile the Linux kernel.

Installing the QLogic Driver — Method Two

Follow these steps to integrate the QLogic driver into the kernel:

1. Change to the directory to which the tar archive was saved.

```
cd /home/qlogic
```

2. Uncompress and extract the source files from the tar archive:

```
tar zxvf qla2x00src-v2.19.15.tgz
```

The files are:

- **Makefile**
- **ql2100_fw.h**
- **ql2200_fw.h**
- **qla2100.h**
- **qla2100.c**
- **release.txt**
- **emc.txt**

3. Copy the firmware and source files to the appropriate directory:

```
cp ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c /usr/src/linux/drivers/scsi
```

4. Once the files have been copied to the appropriate directory, you must edit the kernel as described in the following steps so the driver will be built properly when the kernel is compiled.

Edit Config.in

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

In order for the QLogic driver to be listed as an option in **config/menuconfig/xconfig**, edit **Config.in** as follows:

1. **`cd /usr/src/linux/drivers/scsi`**
2. **`vi Config.in`**
3. If the Linux distribution is Red Hat v6.2, proceed to step 3a. If the distribution is v7.0, go to step 3b:
   a. If the Linux distribution is Red Hat v6.2, search for the following code block around line 126:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'QLogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'QLogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
fi
dep_trisate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Between the `fi` and the `dep_tristate 'Seagate…'` lines, insert the following statement:

```
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
```

The above is one line. It appears here on two lines because of space limitations.

The result should look like the following:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'QLogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'QLogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
fi
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

b. If the Linux distribution is Red Hat v7.0, search for the following code block around line 137:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'Qlogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'Qlogic QLA 1280/12160 SCSI support' CONFIG_SCSI_QLOGIC_1280
   $CONFIG_SCSI
 dep_tristate 'Qlogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
 if [ "$CONFIG_SCSI_QLOGIC_FC" != "y" ]; then
 dep_tristate 'Qlogic QLA 2x00 FC SCSI support' CONFIG_SCSI_QLOGIC_2x00
   $CONFIG_SCSI
 fi
fi
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Between the `fi` and the `dep_tristate 'Seagate…'` lines, insert the following statement:

```
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
```

The above is one line. It appears here on two lines because of space limitations.

The result should look like the following:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'Qlogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'Qlogic QLA 1280/12160 SCSI support' CONFIG_SCSI_QLOGIC_1280
   $CONFIG_SCSI
 dep_tristate 'Qlogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
 if [ "$CONFIG_SCSI_QLOGIC_FC" != "y" ]; then
 dep_tristate 'Qlogic QLA 2x00 FC SCSI support' CONFIG_SCSI_QLOGIC_2x00
   $CONFIG_SCSI
 fi
fi
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Edit Makefile

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

Edit the **Makefile** (located in **/usr/src/linux/drivers/scsi**) as described below.

Note the occurrences of `qla2x00.o`. This applies to UP systems. For SMP systems, these entries are `qla2x00smp.o`.

1. **`vi Makefile`**
2. If the Linux distribution is Red Hat v6.2, proceed to step 2a. If the distribution is v7.0, go to step 2b:
   a. If the Linux distribution is Red Hat v6.2, search for the following code block around line 325:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif

ifeq ($(CONFIG_SCSI_AHA152X),y)
L_OBJS += aha152x.o
else
     ifeq ($(CONFIG_SCSI_AHA152X),m)
     M_OBJS += aha152x.o
     endif
endif
```

   Insert the following lines between the `qlogicfc` and `aha152x` blocks:

```
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
```

The result should look like the following:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_AHA152X),y)
L_OBJS += aha152x.o
else
```

b. If the Linux distribution is Red Hat v7.0, search for the following code block around line 334:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_2x00),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_2x00),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_CPQFCTS,y)
L_OBJS += cpqfc.o
else
     ifeq ($(CONFIG_SCSI_CPQFCTS),m)
     M_OBJS += cpqfc.o
     endif
endif
```

Insert the following lines between the `qla2x00` and `cpfqc` blocks:

```
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
```

The result should look like the following:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_2x00),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_2x00),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_CPQFCTS,y)
L_OBJS += cpqfc.o
else
     ifeq ($(CONFIG_SCSI_CPQFCTS),m)
     M_OBJS += cpqfc.o
     endif
endif
```

3. Around line 703 (if using Red Hat v6.2) or 760 (if using Red Hat v7.0), search for the following code block:

```
megaraid.o: megaraid.c
    $(CC) $(CFLAGS) -c megaraid.c

scsi_mod.o: $(MIX_OBJS) hosts.o scsi.o scsi_ioctl.o constants.o \
        scsicam.o scsi_proc.o scsi_error.o scsi_obsolete.o
           scsi_queue.o
```

Insert the following lines between the `megaraid` and `scsi_mod` sections:

```
qla2x00.o: ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c
        $(CC) $(CFLAGS) -c qla2100.c -o qla2x00.o
```

The result should look like the following:

```
megaraid.o: megaraid.c
    $(CC) $(CFLAGS) -c megaraid.c
qla2x00.o: ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c
        $(CC) $(CFLAGS) -c qla2100.c -o qla2x00.o
scsi_mod.o: $(MIX_OBJS) hosts.o scsi.o scsi_ioctl.o constants.o \
        scsicam.o scsi_proc.o scsi_error.o scsi_obsolete.o
           scsi_queue.o
```

Edit hosts.c

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

Edit **hosts.c** (located in **/usr/src/linux/drivers/scsi**) as follows:

1. **`vi hosts.c`**
2. Around line 190 (if using Red Hat v6.2) or 198 (if using Red Hat v7.0), search for the following code block:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
#include "qlogicfc.h"
#endif

#ifdef CONFIG_SCSI_SEAGATE
#include "seagate.h"
#endif
```

Insert the following lines between the `qlogicfc` and `seagate` sections:

```
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
#include "qla2100.h"
#endif
```

The result should look like the following:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
#include "qlogicfc.h"
#endif

#ifdef CONFIG_SCSI_QLOGIC_QLA2100
#include "qla2100.h"
#endif

#ifdef CONFIG_SCSI_SEAGATE
#include "seagate.h"
#endif
```

3. Around line 515 (if using Red Hat v6.2) or 542 (if using Red Hat v7.0), search for the following code block:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
 QLOGICFC,
#endif
#ifdef CONFIG_SCSI_PAS16
 MV_PAS16,
#endif
```

Insert the following lines in between the `endif` and `ifdef`:

```
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
 QLA2100_LINUX_TEMPLATE,
#endif
```

The result should look like the following:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
 QLOGICFC,
#endif
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
 QLA2100_LINUX_TEMPLATE,
#endif
#ifdef CONFIG_SCSI_PAS16
 MV_PAS16,
#endif
```

Modular Driver Specifics

1. In order to load the driver automatically at boot time, **/etc/conf.modules** must be edited.

   ```
   vi /etc/conf.modules
   ```

   Add the appropriate entry:

System:	Enter:
UP	**alias scsi_hostadapter qla2x00**
SMP	**alias scsi_hostadapter qla2x00smp**

2. Proceed to *Required Kernel Modifications* on page 60 to modify and compile the Linux kernel.

Required Kernel Modifications

Make the following changes to the kernel.

Edit hosts.h

In order to see more than four SCSI hard drives, edit **hosts.h** as follows:

```
cd /usr/src/linux/drivers/scsi
vi hosts.h
```

The text of **hosts.h** differs depending on the Red Hat distribution version:

- v6.2:

  ```
  #define SD_EXTRA_DEVS     40
  #else
  #define SD_EXTRA_DEVS     4   (Change to 128)
  #endif
  ```

- v7.0:

  ```
  #define SD_EXTRA_DEVS     40   (Change to 128)
  #define ST_EXTRA_DEVS     3
  #define SR_EXTRA_DEVS     8
  ```

Edit scsi.c

To allow the host to access and utilize more than eight SCSI LUNs, edit **scsi.c** as follows:

```
cd /usr/src/linux/drivers/scsi
vi scsi.c
```

```
#ifdef CONFIG_SCSI_MULTI_LUN
static int max_scsi_luns = 8;  (Change to 128)
#else
static int max_scsi_luns = 1;
#endif
```

This can also be accomplished by appending a line to **/etc/lilo.conf**:

```
vi /etc/lilo.conf
```

Add this line:

```
append="max_scsi_luns=128"
```

Edit sg.h

The value for the SCSI generic buffer must be increased. Change to the appropriate directory and increase the value for **SG_SCATTER_SZ** to the maximum of **131072**.

```
cd /usr/src/linux/include/scsi
vi sg.h
```

Around line 195, look for the following code block:

```
#define SG_SCATTER_SZ (8 * 4096)  /* PAGE_SIZE not available to user */
/* Largest size (in bytes) a single scatter-gather list element can have.
   The value must be a power of 2 and <= (PAGE_SIZE * 32) [131072 bytes on
   i386]. The minimum value is PAGE_SIZE. If scatter-gather not supported
   by adapter then this value is the largest data block that can be
   read/written by a single scsi command. The user can find the value of
   PAGE_SIZE by calling getpagesize() defined in unistd.h . */
```

Change the value for **SG_SCATTER_SZ (8* 4096)** to (**32 * 4096)**.

After the kernel compilation and system reboot, using the following command can validate this change:

```
cat /proc/sys/kernel/sg-big-buff
```

Proceed to *Configuring and Compiling the Kernel* on page 26 to compile the Linux kernel.

Additional Notes

Dynamic Device Reconfiguration

Linux currently lacks a command built into the kernel that allows for a dynamic SCSI channel reconfiguration like **drvconfig** or **ioscan**. Depending upon the hardware configuration and on whether the QLogic driver has been loaded into the kernel as a module, the QLogic driver can be unloaded and then reloaded. If the QLogic driver has been loaded as a module, this unloading and reloading will cause a scan of the SCSI bus. The driver should detect the newly added device(s). If the QLogic driver is loaded into the kernel statically, the system will need to be rebooted in order for the devices to be detected reliably.

The **modprobe** command is a wrapper or an extension to **insmod**. It uses and maintains a set of files that describe all the modules that are available for the current kernel in **/lib/modules**. modprobe can be used to load and unload an entire set of modules in addition to just a single module. Both the loading and unloading of the QLogic modular driver is done with **modprobe**.

The unloading of the module is done with **modprobe -r qla2x00(smp). modprobe** with the **-r** switch used to unload the loadable modules from the running kernel if they are not in use and if other modules are not dependent upon them.

In the standard Linux v2.2.x kernel, the SCSI addresses are not used in the device names as they are in other types of UNIX (Sun, SGI, HP-UX, and BSD, for example). Block device filenames take the form **/dev/sdln**, where **l** is the letter denoting the physical drive and **n** is the number denoting the partition on that physical drive. Device names and minor numbers are assigned dynamically at boot time or device loading time in the order of discovery.

Depending upon the hardware configuration, if a device is added and the system rebooted, the device numbering can and will change. This change makes the host's mount table inaccurate. For the most consistent results and to reduce the possibility of mount table inaccuracies, new devices should be appended to the list of already attached devices. For example, if the host contains multiple HBAs, it would be best to append the new device to the last HBA and to the end of the device list attached to that HBA. This would eliminate the need to alter the pre-existing entries in the mount table, since the new device could be appended to that as well. If a new device were added to the first out of two HBAs and the system rebooted, the devices would all shift by one number and the mount table entries would also need to be shifted by one device. If there is only one HBA, the

new device can more easily be appended to the list of the regularly attached devices and the mount table altered accordingly.

tasks.h

There is a limit to the number of processes that can run simultaneously on a host. The default is set to 512. If the environment requires it, this value can be increased up to 4092.

To change the value:

1. **cd /usr/src/linux/include/linux**
2. **vi tasks.h**
3. Line 14 is:

```
#define NR_TASKS   512    /*On x86 Max 4092, or 4090 w/ APM configured.*/
```

Change the value of `512` to whatever is required up to a maximum of 4092; then proceed to *Configuring and Compiling the Kernel* on page 26.

BLIST_NOLUN

Hosts with a GEM SCSI chip on the motherboard may incorrectly report multiple processors when booting. This incorrect reporting can be fixed by adding an entry to **BLIST_NOLUN** in the file **scsi.c**.

For example, in the NEC ExpressServer 5800 120Ra2, the GEM chip incorrectly reports back 8 LUNs instead of 1 LUN as it should. This problem can be corrected by editing **scsi.c** as follows:

1. **cd /usr/src/linux/drivers/scsi**
2. **vi scsi.c**
3. Search for the following:

```
   static struct dev_info device_list [] =
{"Aashima", "IMAGERY 2400sp", "1.03", BLIST_NOLUN}, /*Locks up if
   polled for lun != 0 */
{"CHINON", "CD-ROM CDS-431", "H42", BLIST_NOLUN}, /*Locks up if polled
   for lun != 0*/
```

where:

- `Aashima` = Make
- `IMAGERY2 2400sp` = Model
- `1.03` = firmware
- `BLIST_NOLUN` = flag

4. Go to the end of the `BLIST_NOLUN` list and add the device. For example, in the case of the NEC ExpressServer 5800 120Ra2, the entry would be similar to the following:

```
{"NEC", "GEM312R2-G7CGQ", "4.1a", BLIST_NOLUN}, /*Reports back
   incorrect LUNs*/
```

5. After this addition is made, proceed to *Configuring and Compiling the Kernel*.

Configuring and Compiling the Kernel

If this is a new installation, proceed to *New Installation* later on this page. If this is an upgrade to a newer kernel revision, go to *Kernel Upgrade* on page 29.

New Installation

Follow these steps to configure the host in a new installation:

1. Prior to configuring the kernel, change to the appropriate directory:

   `cd /usr/src/linux`

2. Configure the kernel:

   `make config/menuconfig/xconfig`

3. From the main menu, select the **Processor type and features** menu.

   a. Select the appropriate **Processor family** for the processors in the system:

      – 386
      – 486/Cx486
      – 586/K5/5x86/6x86
      – Pentium/K6/TSC
      – PPro/6x86MX/PII
      – PIII/Xeon/Deschutes

   b. Enable **Symmetric multi-processing support** as static ***only*** if the system being used has multiple processors.

   c. Select **<Exit>** to return to the Main Menu.

4. From the main menu, select the **SCSI support** menu (Figure 2 on page -27).

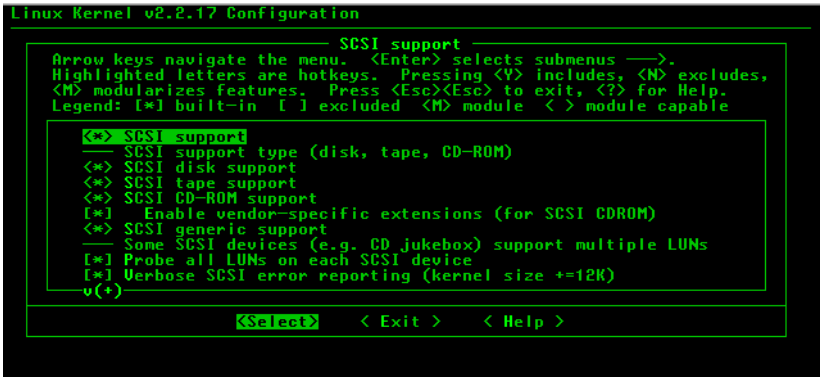
Linux Kernel v2.2.17 Configuration
SCSI support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> SCSI support
--- SCSI support type (disk, tape, CD-ROM)
<*> SCSI disk support
<*> SCSI tape support
<*> SCSI CD-ROM support
[*] Enable vendor-specific extensions (for SCSI CDROM)
<*> SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*] Probe all LUNs on each SCSI device
[*] Verbose SCSI error reporting (kernel size +=12K)
v(+)
<Select> < Exit > < Help >

Figure 2 **SCSI Support Menu**

Select **SCSI support** as static (* or **y**).

Under

---SCSI support type (disk, tape, CD-ROM)

select the options **SCSI disk support** and **SCSI generic support** as static (* or **y**).

Under

---Some SCSI devices (e.g. CD jukebox) support multiple LUNs

select **Probe all LUNs on each SCSI device** as static (* or **y**).

The **qlogicfc** driver must be disabled in the kernel to eliminate any possible conflicts between the **qlogicfc** and **qla2x00(smp)** drivers.

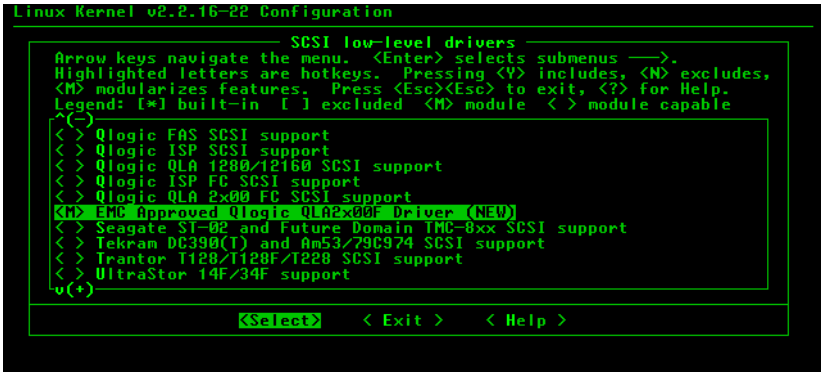
Linux Kernel v2.2.16-22 Configuration
SCSI low-level drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
^(-)
< > Qlogic FAS SCSI support
< > Qlogic ISP SCSI support
< > Qlogic QLA 1280/12160 SCSI support
< > Qlogic ISP FC SCSI support
< > Qlogic QLA 2x00 FC SCSI support
<M> EMC Approved Qlogic QLA2x00F Driver (NEW)
< > Seagate ST-02 and Future Domain TMC-8xx SCSI support
< > Tekram DC390(T) and Am53/79C974 SCSI support
< > Trantor T128/T128F/T228 SCSI support
< > UltraStor 14F/34F support
v(+)
<Select> < Exit > < Help >

Figure 1-1 **SCSI Low-Level Drivers Menu**

Under

SCSI low-level drivers

deselect **QLogic ISP FC SCSI support**.

deselect **Qlogic QLA 2x00 FC SCSI support**.

If Method Two was used to incorporate the QLogic driver into the kernel (refer to *Installation Methods* on page 10), you must select the QLogic driver in **menuconfig** as static or modular.

Under

SCSI low-level drivers

select **EMC-approved QLogic QLA2x00F Driver** as modular (**M** or **m**) or static (***** or **y**), as required. (Refer to the above note.)

Select **Exit** to return to the Main Menu.

From the Main Menu, select **Exit**; then select **Yes** to save your configuration in response to the question `Do you wish to save your new kernel configuration?`

The configuration will be saved to the file **/usr/src/linux/.config**.

5. Compile the kernel:

```
make dep
make clean
make install
make modules
make modules_install
cd /boot
```

6. Build a ramdisk image:

```
mkinitrd /boot/initrd-$1.img $1
```
(where **$1** is the kernel version)

For example:

```
mkinitrd /boot/initrd-2.2.17.img 2.2.17
```

The **mkinitrd** command is used to create a new ramdisk image and is dependent upon the loopback device being available. If the loopback driver is selected in the kernel as a module, ensure that it is loaded with **#lsmod**. If the loopback driver was not loaded, insert it with **#insmod loop**.

---

7. Run **lilo** to update the loading map and reboot:

```
lilo -v
reboot
```

Kernel Upgrade

For a kernel upgrade, install Linux from the Red Hat v6.2 CD, following the procedure in the installation guide. Partition the boot drive and select the packages and services necessary for the host. The host can be upgraded from Red Hat v6.2 or Red Hat v7.0 to up to v2.2.17 of the Linux kernel.

To upgrade the host to a newer version of the kernel, the kernel must be downloaded, from: **http://www.kernel.org/mirrors/**.

1. Prepare the host for the new version of the kernel:

```
cd /usr/src
rm -f linux
```

2. Untar the kernel and create a directory for it:

```
tar zxvf linux-$1.tar.gz
mv linux linux-$1
ln -s linux-$1 linux
cd linux
```
(where **$1** is the kernel version)

3. Delete stale **.o** files and dependencies:

   ```
   make mrproper
   ```

4. Configure the kernel:

   ```
   make config/menuconfig/xconfig
   ```

   From the main menu, select the **SCSI support** menu (Figure 4):

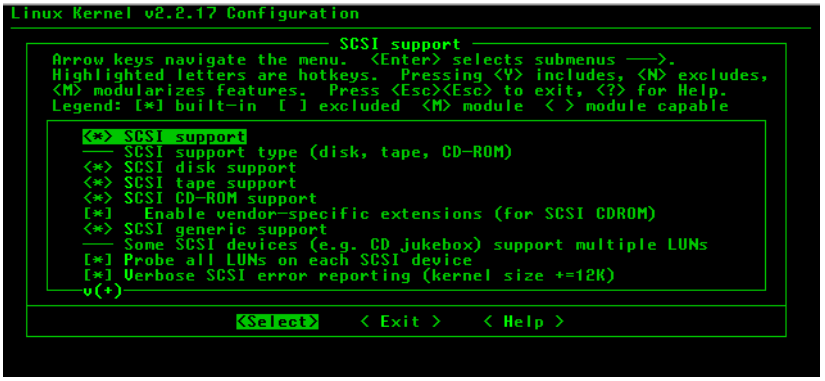
Linux Kernel v2.2.17 Configuration
SCSI support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> SCSI support
--- SCSI support type (disk, tape, CD-ROM)
<*> SCSI disk support
<*> SCSI tape support
<*> SCSI CD-ROM support
[*] Enable vendor-specific extensions (for SCSI CDROM)
<*> SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*] Probe all LUNs on each SCSI device
[*] Verbose SCSI error reporting (kernel size +=12K)
v(+)
<Select> < Exit > < Help >

Figure 1-2 **SCSI Support Menu**

Select **SCSI support** as static (* or **y**).

Under

---SCSI support type (disk, tape, CD-ROM)

select the options **SCSI disk support** and **SCSI generic support** as static (* or **y**).

Under

---Some SCSI devices (e.g. CD jukebox) support multiple LUNs

select **Probe all LUNs on each SCSI device** as static (* or **y**).

The **qlogicfc** driver must be disabled in the kernel to eliminate any possible conflicts between the **qlogicfc** and **qla2x00(smp)** drivers.

Under

SCSI low-level drivers

deselect **QLogic ISP FC SCSI support**.

deselect **Qlogic QLA 2x00 FC SCSI support**.

If Method Two was used to incorporate the QLogic driver into the kernel (refer to *Installation Methods* on page 10), you must select the QLogic driver in **menuconfig** as static or modular.

Under

SCSI low-level drivers

select **EMC-approved QLogic QLA2x00F Driver** as modular (**M** or **m**) or static (***** or **y**), as required. (Refer to the above note.)

5. Compile the kernel:

```
make dep
make clean
make install
make modules
make modules_install
cd /boot
```

6. Add the new kernel version to **lilo.conf** so it will be bootable:

```
vi /etc/lilo.conf
```

For example:

```
image=/boot/vmlinuz-2.2.17
label=linux-2.2.16
root=/dev/sda1
initrd=/boot/initrd-2.2.17.img
read-only
```

7. Build a ramdisk image:

```
mkinitrd /boot/initrd-$1.img $1
```
(where **$1** is the kernel version)

For example:

```
mkinitrd /boot/initrd-2.2.17.img 2.2.17
```

The **mkinitrd** command is used to create a new ramdisk image and is dependent upon the loopback device being available. If the loopback driver is selected in the kernel as a module, ensure that it is loaded with **#lsmod**. If the loopback driver was not loaded, insert it with **#insmod loop**.

8. Run **lilo** to update the loading map and reboot:

```
lilo -v
reboot
```

Incorporating Symmetrix Fibre Channel

Once the Symmetrix has devices assigned with device addresses and the directors have been switched on line, the host can see all the target Symmetrix devices assigned to that host interface. Devices are presented to the host in the same manner as devices accessed through a standard SCSI interface.

Partitioning and Labeling a New Device

Block device filenames take the form **/dev/sd*ln***, where ***l*** is the letter denoting the physical drive and ***n*** is the number denoting the partition on that physical drive. Usually, the partition number is not included when the device corresponds to the entire drive.

Following this format, the filenames would appear as follows:

/dev/sd[a-d][a-z][1-15]

Linux supports a total of 128 SCSI devices per system. As a result, the host could support disk devices from **/dev/sda** through **/dev/sddx**.

Linux supports one to 15 partitions per disk device. Partitions 1 through 4 are the primary partitions, while partitions 5 and greater are the logical or extended partitions. (These limitations are specific to the Intel platform.) By default, slices are not used in Linux.

True raw devices have not yet been incorporated into the kernel.

The corresponding character device filenames take the form **/dev/sg[a-d][a-z]**, where **sg** is one of four high-level device drivers in the SCSI subsystem.

The four high-level device drivers are:

sg (SCSI generic interface)

sd (direct access; disks, for example)

st (tapes)

sr (data CD-ROMs)

The **sg** driver is a character-based device, while the other three drivers are block-based devices. **sg** is used primarily for scanners, CD writers, printers, and raw access, although **sg** is not a true raw device. The **sg** device files are dynamically mapped to SCSI IDs/LUNs on the SCSI bus, starting with the first SCSI controller.

The Linux kernel assigns minor SCSI device numbers dynamically by assigning them only to devices that are actually connected to the host in the order of their SCSI IDs. This means that connecting an external SCSI device can change the minor numbers of all the internal SCSI devices with a higher SCSI ID.

Partitioning Symmetrix Devices

Once the QLogic driver is loaded, the Symmetrix devices will be represented in **/proc/scsi/scsi**. They can be viewed by using **more /proc/scsi/scsi**. Only **/dev/sda** through **/dev/sdp** and **/dev/sga** through **/dev/sgh** will be created by default upon kernel compilation. In order to create subsequent device files for the Symmetrix devices attached to the host, the root user will need to create them via the **mknod** command.

Using the **fdisk** command allows multiple partitions to reside on a single disk device.

To partition one or more Symmetrix devices for the use of Linux, log in as *root* and follow the procedure below:

1. Type **`fdisk /dev/sdb`** ENTER.
2. If you want to display a list of options, type **`m`** ENTER (for help) to display the following:

```
Command action
 a     toggle a bootable flag
 b     edit bsd disklabel
 c     toggle the DOS compatibility flag
 d     delete a partition
 l     list known partition types
 m     print this menu
```

```
Command action
 n      add a new partition
 o      create a new empty DOS partition table
 p      print the partition table
 q      quit without saving changes
 s      create a new empty Sun disklabel
 t      change a partition's system ID
 u      change display/entry units
 v      verify the partition table
 w      write table to disk and exit
 x      extra functionality (experts only)
```

To display a list of advanced options, type **x** ENTER (for expert functionality), then **m** ENTER to display the following:

```
Command action
 b      move beginning of data in a partition
 c      change number of cylinders
 d      print the raw data in the partition table
 e      list extended partitions
 g      create an IRIX partition table
 h      change number of heads
 m      print this menu
 p      print the partition table
 q      quit without saving changes
 r      return to main menu
 s      change number of sectors
 v      verify the partition table
 w      write table to disk and exit
```

If you display the above advanced menu, type **m** ENTER when finished viewing the menu, to return to the main menu.

3. On the main menu, select the options you need to create and label a new partition:
   - Type **n** ENTER to add a new partition.
   - Type **t** ENTER to change a partition's system ID.
   - You can type **L** ENTER for a list of hex codes for system IDs. (Linux is type 83. Linux swap is type 82.)
   - Type **p** ENTER to print the partition table and to verify that the selections look correct.
   - Type **w** ENTER to write the table to disk and exit **fdisk**.

4. Repeat step 3 for each new Symmetrix device.

cfdisk is a curses-based disk partition table manipulator for Linux that is used instead of **fdisk**.) If used with the **-z** option, **cfdisk** allows the device's partition table to be zeroed out. The use of **cfdisk -z** is required if you are converting a device previously used by another host.

Example: **`cfdisk -z /dev/sdb`**

The resulting display would be similar to the following:

```
                    cfdisk 0.8n

                 Disk Drive: /dev/sdb
       Heads: 255   Sectors per Track: 63   Cylinders: 550

  Name      Flags     Part Type     FS Type     [Label]     Size (MB)
------------------------------------------------------------------------

               Pri/Log  Free Space              4314.34




  [  Help  ]  [  New   ]  [ Print  ]  [  Quit  ]  [ Units  ]
  [ Write  ]
```

CAUTION

Changing the partition table results in loss of any previous data.

Select **[New]**. The following options appear at the bottom of the window:

[Primary] [Logical] [Cancel]

Select **[Primary]** to create a primary Linux partition and select the size of the partition that is required.

In this case, to use the entire device, select:

Size (in MB): 4314.34

Once the above step is completed, the options listed at the bottom will change to the following:

```
                        cfdisk 0.8n

                     Disk Drive: /dev/sdb
          Heads: 255   Sectors per Track: 63   Cylinders: 550

   Name      Flags     Part Type     FS Type     [Label]     Size (MB)
------------------------------------------------------------------------

   sdb1                Primary      Linux                   4314.34




  [  Bootable  ]  [  Delete   ]  [ Help  ]  [  Maximize  ]  [ Print  ]
  [ Quit  ] [  Type  ][  Units  ] [  Write  ]
```

cfdisk will automatically select Linux as the file system type. If a different type of file system or a swap partition is required, select **[Type]** and choose the necessary file system type.

Select **[Write]** to write the partition table to the disk. **cfdisk** will prompt the user with:

```
"Are you sure you want to write the partition table to disk (yes or no)?
   Warning!! This may destroy data on your disk!"
```

Type **yes** ENTER to write the partition table. Once the partition table has been written to the disk, select **[Quit]** to exit **cfdisk**.

Inaccessible HBA Fix

In the event that the HBA becomes inaccessible, follow the steps listed below:

1. Unplug the fiber cables from the host.
2. Reboot the host.
3. Upon rebooting, press ALT-Q when prompted to enter the **Fast!Util** menu.
4. From the **Fast!Util Options** menu, select **Configuration Settings**.
5. On the **Configuration Settings** menu, select **Restore Default Settings**.
6. Return to the procedure under *Setting the NVRAM Variables* on page 7 to reestablish the EMC-recommended adapter settings.

7. Save the settings for the HBA installed in the host and exit **Fast!Util**.
8. Plug the fiber cables back into the HBA(s) in the host and reboot.
9. Upon rebooting, press ALT-Q when prompted to enter the **Fast!Util** menu.
10. From the **Fast!Util Options** menu, select **Fibre Disk Utility**.
11. Follow the instructions from *Verifying the Disk Media* on page 9 to ensure that the HBA has been reset and can now access all of the Symmetrix devices.
12. Exit out of the Fibre Disk utility and reboot the host.
13. Once the Symmetrix devices are partitioned and labeled appropriately, proceed with *Creating and Mounting a File System*.

Creating and Mounting a File System

To create standard UNIX file systems under Linux, log in as *root* and proceed as follows for each previously labeled device.

Create a New File System

Once each Symmetrix disk device has been formatted, partitioned, and labeled, a file system needs to be created in order to make the partition usable. Use the **mkfs** command to create a file system.

For a single partition, use **`mkfs /dev/sdb1`**.

You will be prompted to confirm the construction of a new file system. Type **`Y`** ENTER.

Create a Mount Directory

Once the file system for each Symmetrix device has been created, a mount directory for each device must be created.

To create a mount directory, use **`mkdir /mnt/sdb`**, where **`/mnt/sdb`** is the complete path for the new file system.

By default, **/mnt** is created upon installation, but a separate directory can be created in which to mount the devices' file systems.

Mount the File System

To mount each file system, enter a statement similar to the following:

```
mount -t ext2 /dev/sdb1 /mnt/sdb
```

This will mount the raw device **/dev/sdb1** onto the mount point **/mnt/sdb**.

System and Error Messages

Linux logs system and error messages to a file called **/var/log/messages**. This file includes SCSI error messages. Linux also logs boot messages in a file called **/var/log/boot.log**. The host will also display these messages at the system console.

Fabric Configurations

This section describes the procedures required to install one or more EMC-approved Qlogic host bus adapters into a Linux host and configure the host for Symmetrix Fibre Channel fabric.

Useful Linux Utilities and Functions

These utilities and functions can be helpful in performing configuration operations:

Utility/Function	Description
fdisk	Command used to create and manipulate Linux partition tables
cfdisk	Curses-based disk partition table manipulator for Linux
fsck	Command used to check and repair a Linux file system
mkfs	Command used to create Linux file system on a device
mount	Command used to attach the file system on a device to the file tree
umount	Command used to detach a file system
shutdown	Command used to shut down the system gracefully
reboot	Command used to stop and restart the operating system
modprobe	Utility used to load or remove a set of modules that can be either a single module or a stack of dependent modules

Installing the HBA

Before the HBA is installed, the host must be configured with Linux. Install Linux from the RedHat v6.2 CD, following the procedure provided in the RedHat installation guide. Partition the boot drive and select the packages and services necessary for the host.

To install the QLA2200F, follow the instructions included with your adapter. The adapter installs into a single slot, and has no user-configurable jumpers or switches.

Configuring Firmware Settings

After the HBA is installed, follow these steps to verify and configure HBA firmware settings.

Verifying the Correct Firmware Version

The firmware (HBA BIOS) version can be determined at boot time from the QLA2200F banner or from the **Fast!Util Options** menu.

- To determine the firmware version from the QLA2200F banner, boot the host and watch for the banner:

```
Qlogic Corporation
QLA2200 PCI Fibre Channel ROM BIOS Version 1.61
Copyright © Qlogic Corporation 1993-1999 All rights reserved
Press <ALT - Q> for Fast!UTIL
www.qlogic.com
```

Figure 2 **QLA2200F Banner**

  - If the ROM BIOS is v1.61, continue to *Setting the NVRAM Variables* on page 42.
  - If the ROM BIOS is not v1.61, upgrade the firmware as described under *Upgrading the HBA BIOS* on page 41; then go to *Setting the NVRAM Variables* on page 42.

- To determine the firmware version from the QLA2200 **Fast!Util Options** menu:

  a. Boot the host. As soon as the QLA2200F banner (see Figure 2) appears, press ALT-Q to enter the **Fast!Util Options** menu.

  b. Select **Configuration Settings** from the menu.

  c. Select **Host Adapter Settings** from the **Configuration Settings** menu.

  d. Under **Host Adapter Settings**, note the **BIOS Version**:

  - If the version is 1.61, continue to *Setting the NVRAM Variables*.
  - If the version is not 1.61, upgrade the firmware as described under *Upgrading the HBA BIOS* on page 41; then proceed to *Setting the NVRAM Variables*.

Upgrading the HBA BIOS

Follow these steps to upgrade the HBA BIOS:

1. Obtain the latest version of the BIOS from the QLogic website:
   a. Access **http://www.qlogic.com.**
   b. Click **Driver Download.**
   c. Scroll to the bottom of the QLogic Drivers License Agreement and click **Use QLogic Drivers/Software**.
   d. Click **EMC Approved Drivers**.
   e. Click **EMC Approved QLA22xx Drivers**.
   f. Click **Link to ROM BIOS** in the **ROM BIOS v1.61** section.
   g. Save the file to a local directory as **22emc03bios161.exe**.
2. The file is a self-extracting ZIP file that must be extracted to a bootable diskette drive, as follows:
   a. Insert a diskette into a Microsoft Windows machine.
   b. Open any DOS window.
   c. At the DOS prompt, format the diskette:

      ```
      format /s a:
      ```

   d. At the DOS prompt, change (`cd`) to the directory in which you saved the zipped file, then extract the file to the diskette:

      ```
      22emc03bios161 a:
      ```

The QLA2200-EMC HBA must be installed in the host before proceeding.

---

3. Insert the BIOS upgrade installation diskette into the diskette drive.
4. Reboot the host.
5. After the host has rebooted, a DOS prompt appears. Type **`flasutil /L /F`** ENTER.

The HBA BIOS upgrade might take a few minutes.

---

6. After the upgrade is complete, remove the diskette and reboot the host. During boot-up, the QLA2200F banner (see Figure 2 on page -40) should display BIOS Version 1.61.

Setting the NVRAM Variables

Follow the steps below to set the NVRAM variables.

CAUTION

Any time the default settings for the adapter are chosen via Fast!Util, the NVRAM settings specific to EMC are overwritten. EMC recommends that you do not select the default adapter settings option from Fast!Util.

1. As the host boots, press ALT-Q when prompted `Press <Alt-Q> for Fast!Util`.
2. After the **Fast!Util** program loads, the display will depend on whether there are multiple QLogic HBAs installed:
   - If there is only one QLogic HBA, a **Fast!Util Options** menu will appear.
   - If there are multiple QLogic HBAs, a list of addresses occupied by those HBAs will appear. Using the arrow keys, select the desired HBA and press ENTER. The **Fast!Util Options** menu will then appear.
3. From the **Fast!Util Options** menu, select **Configuration Settings.**
4. On the **Configuration Settings** menu, select **Host Adapter Settings**.
5. Change the setting for **Frame Size** to **2048**.
6. Select **Host Adapter BIOS** and press ENTER to change the setting to **Disabled**.
7. Select **Adapter Hard Loop ID** and press ENTER to change the setting to **Disabled**.
8. Press ESC to return to the **Configuration Settings** menu.
9. Select **Selectable Boot Settings** from the **Configuration Settings** menu. Make sure that **Selectable Boot Device** is **Disabled**. If it is not, press ENTER to change the setting.
10. Press ESC to return to the **Configuration Settings** menu.
11. On the **Configuration Settings** menu, select **Advanced Adapter Settings**.
12. Increase the setting for **Execution Throttle** to **256**.

13. Change the setting for **LUNs per Target** to **128**. (The factory setting is **8**, but this must be changed to utilize all 128 LUNs available from the Symmetrix.)
14. Select **Enable LIP Full Login** and press ENTER to change the setting to **Yes**.
15. Select **Enable Target Reset** and press ENTER to change the setting to **Yes.**
16. Increase the setting for **Port Down Retry Count** to **32**.
17. Press ESC to return to the **Configuration Settings** menu.
18. On the **Configuration Settings** menu, select **Extended Firmware Settings**.
19. Change the setting for **Connection Options** to **1** (Point to Point only).
20. Press ESC to return to the **Fast!Util Options** menu.
21. When prompted to save the changes made to the current adapter, select **Save Changes**.
22. If you have more adapters to configure, choose **Select Host Adapter** and repeat steps 4 through 21 for each adapter.
23. Use the Fibre Disk utility to verify the disk media, as described under *Verifying the Disk Media* on page 44.
24. Press ESC to exit **Fast!Util**.
25. Reboot the host.

Verifying the Disk Media

To verify that the HBA can see the Symmetrix, use the Fibre Disk utility to scan and verify disk media.

1. Under the **Fast!UTIL Options** menu, select **FIBRE DISK UTILTY**.
2. You should see `SYMMETRIX` in a Fibre Channel device list of all possible 128 targets, as shown below. Page down if necessary to display all targets in the list.

```
   ID              VENDOR    PRODUCT     REV                     NODE NAME
<Hard loop ID>     EMC       SYMMETRIX   <micro code revision>   01000006048___
```

   - If `SYMMETRIX` appears in the list, go to step 3.
   - If `SYMMETRIX` does not appear in the list, verify the following, then start this procedure again at step 1:
     - that the correct Symmetrix port is connected to the host.
     - that the Symmetrix is on line.
     - that the switch is on line.
     - that zoning is set up on the switch.
     - that the cables are not damaged and are plugged in.
3. Use the arrow keys to select **SYMMETRIX** from the list of targets, then press ENTER.
4. A list of LUNs appears—select any LUN from the list.
5. A **Disk Utility Options** menu appears. Select **VERIFY DISK MEDIA**.
6. Select **CONTINUE WITH VERIFY**.
7. After media verification has completed:
   - If the message `Media Verification Complete` appears, press ESC repeatedly to exit **Fast!Util** and reboot the host, then continue with *Configuring the Host with the QLogic HBA Driver* on page 45.
   - If the media verification fails, verify that the cables and connectors are not damaged.

Configuring the Host with the QLogic HBA Driver

Using the QLogic adapter with the Linux operating system requires HBA driver software. The driver functions at a layer below the Linux SCSI driver to present Fibre Channel devices to the operating system as if they were standard SCSI devices.

Installation Methods

The QLogic driver can be installed onto a Linux host using one of two methods:

- Method One — Build the driver in the directory created for it.
- Method Two — Integrate the driver into the kernel.

If the driver is to be statically built in the kernel, the driver must be integrated into the kernel using Method Two.

There are advantages and disadvantages for each method of driver installation.

- Method One — Requires fewer file edits; however, Method One does restrict the use of the driver to modular only. The kernel must still be configured and compiled.
- Method Two — Requires additional edits to integrate the driver into the kernel and to incorporate the driver into **menuconfig**. Method Two is more flexible because it offers the ability to load the driver either as static or modular.

Both methods require editing **hosts.h** and **scsi.c**, and enabling **Probe All LUNs on each SCSI device** in **menuconfig**. Both methods require that the kernel be compiled.

Download the latest driver from the QLogic Web site (refer to *Downloading the Driver* on page 47 later on this page), then select one of the two methods to install the driver onto the Linux host:

- For a modular driver only, follow the procedure under *Installing the QLogic Driver — Method One* on page 48.
- To integrate the QLogic driver into the kernel so that it may be either modular or static, follow the procedure under *Installing the QLogic Driver — Method Two* on page 50.

Discussion of Default and EMC-Approved Drivers

EMC supports *only* the QLogic **qla2x00(smp)** driver, which can be downloaded from the QLogic Website.

EMC *does not* support the **qlogicfc** driver that is included in the Linux kernel (because the driver does not support persistent binding). Also, EMC *does not* support the alternate **qla2x00** driver that is included in the kernel in the RedHat v7.0 distribution.

If the QLA2200F HBA is present in the system prior to a new installation of Linux, the RedHat installation application automatically loads its version of the driver, which will be installed and inserted automatically when the host reboots. For this reason, EMC highly recommends installing the Linux operating system *before* installing the QLogic HBAs.

Removing the Default Driver

If the Qlogic QLA2200F is installed in the system prior to the Linux installation, the default driver *must* be removed from the kernel after rebooting, as follows:

1. Issue the appropriate command:

Linux Version	Enter:
6.2	`modprobe -r qlogicfc`
7.0	`modprobe -r qla2x00`

2. Edit the modules file to remove the unsupported driver:
   a. Issue the appropriate command:

Linux Version	Enter:
6.2	`vi /etc/conf.modules`
7.0	`vi /etc/modules.conf`

   b. Remove the appropriate entry:

Linux Version	Remove:
6.2	`alias scsi_hostadapter qlogicfc`
7.0	`alias scsi_hostadapter qla2x00`

Be aware that if the **qlogicfc** driver is alternately loaded and unloaded with the EMC-Approved QLogic **qla2x00(smp)** driver, there may be problems seeing Symmetrix devices. The two drivers use different firmware revisions; therefore, alternately loading both drivers might have an adverse effect on the HBA.

Downloading the Driver

Use the following procedure to download the QLogic driver from the QLogic Website:

1. Create a directory to which you will save the driver:

   ```
   makedir /home/qlogic
   ```

2. Use a Web browser to access **http://www/qlogic.com.**
3. On the QLogic home page, click **Driver Download.**
4. Scroll to the bottom of the QLogic Drivers License Agreement and click **Use QLogic Drivers/Software**.
5. Click **EMC Approved Drivers**.
6. Click **EMC Approved QLA22xx Drivers**.
7. Click **Link to Source files** in the **Linux driver v2.19.15** section.
8. Save the driver to a local directory you created.

Select one of the two methods to install the driver onto the Linux host:

- For a modular driver only, follow the procedure under *Installing the QLogic Driver — Method One* on page 48.
- To integrate the QLogic driver into the kernel so that it may be either modular or static, follow the procedure under *Installing the QLogic Driver — Method Two* on page 50.

Installing the QLogic Driver — Method One

Follow these steps to configure the driver as modular only:

1. Change to the directory to which the tar archive was saved.

   ```
   cd /home/qlogic
   ```

2. Uncompress and extract the source files from the tar archive:

   ```
   tar zxvf qla2x00src-v2.19.15.tgz
   ```

   The files are:

   - **Makefile**
   - **ql2100_fw.h**
   - **ql2200_fw.h**
   - **qla2100.h**
   - **qla2100.c**
   - **release.txt**
   - **emc.txt**

3. Create the object file and install it into the appropriate directory of **/lib/modules/$1/scsi** (where **$1** is the kernel version).

   To build the driver, enter the appropriate command:

System:	Enter:
UP	`make install`
SMP	`make install SMP=1`

4. To load the QLogic driver on the system, enter the appropriate command:

System:	Enter:
UP	`modprobe qla2x00`
SMP	`modprobe qla2x00smp`

5. In order to load the driver automatically at boot time, **/etc/conf.modules** must be edited:

   ```
   vi /etc/conf.modules
   ```

   Add the appropriate entry:

System:	Enter:
UP	`alias scsi_hostadapter qla2x00`
SMP	`alias scsi_hostadapter qla2x00smp`

6. In a fabric environment, persistent binding *must* be enabled. Without a persistent binding mechanism, the host will be unable to maintain persistent logical routing of the communication from a storage device object across the fabric to a Symmetrix volume. If the physical configuration of the switch is changed (for example, cables are swapped or the host is rebooted), the logical route becomes inconsistent, which could cause possible data corruption.

   To enable persistent binding for the modular driver, you must edit **/etc/conf.modules**. The initiator and target information necessary for persistent binding are logged in **/var/log/messages** when the driver loads. ***grep*** the file to obtain the adapter and target information:

   ```
   grep scsi-qla /var/log/messages
   ```

   The output will be similar to the following:

```
Jan 9 11:24:48 denali kernel: scsi-qla0-adapter-node=0000000000000000;
Jan 9 11:24:48 denali kernel: scsi-qla0-adapter-port=200000e08b009209;
Jan 9 11:24:48 denali kernel: scsi-qla0-target-0=500604800000000c;
```

   The above information is appended to **/etc/conf.modules**. Use the **ql2xopts** parameter to add the persistent binding information to the file. An example of the **/etc/conf.modules** configuration file for a uni-processor system would be:

```
alias scsi_hostadapter aic7xxx
alias scsi_hostadapter qla2x00

options qla2x00 ql2xopts=scsi-qla0-node=0000000000000000\;scsi-qla0-
adapter-port=200000e08b009209\;scsi-qla0-target-0=500604800000000c\;
```

   When configuring and manually loading the **qla2x00(smp)** module with persistent binding information, you must use **modprobe** rather than **insmod**. **modprobe** is a wrapper that is an extension to **insmod**. This utility will load the driver and automatically read the **/etc/conf.modules** configuration file.

7. Run the following command to update your **/lib/modules/$1/modules.dep** file (where **$1** is the kernel version) with the information you added to **/etc/conf.modules**:

   ```
   depmod -a
   ```

8. Proceed to *Required Kernel Modifications* on page 60 to modify and compile the Linux kernel.

Installing the QLogic Driver — Method Two

Follow these steps to integrate the QLogic driver into the kernel:

1. Change to the directory to which the tar archive was saved.

   ```
   cd /home/qlogic
   ```

2. Uncompress and extract the source files from the tar archive:

   ```
   tar zxvf qla2x00src-v2.19.15.tgz
   ```

   The files are:

   - **Makefile**
   - **ql2100_fw.h**
   - **ql2200_fw.h**
   - **qla2100.h**
   - **qla2100.c**
   - **release.txt**
   - **emc.txt**

3. Copy the firmware and source files to the appropriate directory:

```
cp ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c /usr/src/linux/drivers/scsi
```

4. Once the files have been copied to the appropriate directory, you must edit the kernel as described in the following steps so the driver will be built properly when the kernel is compiled.

Edit Config.in

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

In order for the QLogic driver to be listed as an option in **config/menuconfig/xconfig**, edit **Config.in** as follows:

1. `cd /usr/src/linux/drivers/scsi`
2. `vi Config.in`
3. If the Linux distribution is RedHat v6.2, proceed to step 3a. If the distribution is v7.0, go to step 3b:

   a. If the Linux distribution is RedHat v6.2, search for the following code block around line 126:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'QLogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'QLogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
fi
dep_trisate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Between the `fi` and the `dep_tristate 'Seagate…'` lines, insert the following statement:

```
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
```

The above is one line. It appears here on two lines because of space limitations.

The result should look like the following:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'QLogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'QLogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
fi
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

b. If the Linux distribution is RedHat v7.0, search for the following code block around line 137:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'Qlogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'Qlogic QLA 1280/12160 SCSI support' CONFIG_SCSI_QLOGIC_1280
   $CONFIG_SCSI
 dep_tristate 'Qlogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
 if [ "$CONFIG_SCSI_QLOGIC_FC" != "y" ]; then
 dep_tristate 'Qlogic QLA 2x00 FC SCSI support' CONFIG_SCSI_QLOGIC_2x00
   $CONFIG_SCSI
 fi
fi
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Between the `fi` and the `dep_tristate 'Seagate…'` lines, insert the following statement:

```
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
```

The above is one line. It appears here on two lines because of space limitations.

The result should look like the following:

```
if [ "$CONFIG_PCI" = "y" ]; then
 dep_tristate 'Qlogic ISP SCSI support' CONFIG_SCSI_QLOGIC_ISP $CONFIG_SCSI
 dep_tristate 'Qlogic QLA 1280/12160 SCSI support' CONFIG_SCSI_QLOGIC_1280
   $CONFIG_SCSI
 dep_tristate 'Qlogic ISP FC SCSI support' CONFIG_SCSI_QLOGIC_FC $CONFIG_SCSI
 if [ "$CONFIG_SCSI_QLOGIC_FC" != "y" ]; then
 dep_tristate 'Qlogic QLA 2x00 FC SCSI support' CONFIG_SCSI_QLOGIC_2x00
   $CONFIG_SCSI
 fi
fi
dep_tristate 'EMC-approved QLogic QLA2X00F Driver' CONFIG_SCSI_QLOGIC_QLA2100
   $CONFIG_SCSI
dep_tristate 'Seagate ST-02 and Future Domain TMC-8xx SCSI support'
   CONFIG_SCSI_SEAGATE $CONFIG_SCSI
```

Edit Makefile

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

Edit the **Makefile** (located in **/usr/src/linux/drivers/scsi**) as described below.

Note the occurrences of `qla2x00.o`. This applies to UP systems. For SMP systems, these entries are `qla2x00smp.o`.

1. **`vi Makefile`**
2. If the Linux distribution is RedHat v6.2, proceed to step 2a. If the distribution is v7.0, go to step 2b:
   a. If the Linux distribution is RedHat v6.2, search for the following code block around line 325:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif

ifeq ($(CONFIG_SCSI_AHA152X),y)
L_OBJS += aha152x.o
else
     ifeq ($(CONFIG_SCSI_AHA152X),m)
     M_OBJS += aha152x.o
     endif
endif
```

   Insert the following lines between the `qlogicfc` and `aha152x` blocks:

```
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
```

The result should look like the following:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_AHA152X),y)
L_OBJS += aha152x.o
else
```

b. If the Linux distribution is RedHat v7.0, search for the following code block around line 334:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_2x00),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_2x00),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_CPQFCTS,y)
L_OBJS += cpqfc.o
else
     ifeq ($(CONFIG_SCSI_CPQFCTS),m)
     M_OBJS += cpqfc.o
     endif
endif
```

Insert the following lines between the `qla2x00` and `cpfqc` blocks:

```
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
```

The result should look like the following:

```
ifeq ($(CONFIG_SCSI_QLOGIC_FC),y)
L_OBJS += qlogicfc.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_FC),m)
     M_OBJS += qlogicfc.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_2x00),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_2x00),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),y)
L_OBJS += qla2x00.o
else
     ifeq ($(CONFIG_SCSI_QLOGIC_QLA2100),m)
     M_OBJS += qla2x00.o
     endif
endif
ifeq ($(CONFIG_SCSI_CPQFCTS,y)
L_OBJS += cpqfc.o
else
     ifeq ($(CONFIG_SCSI_CPQFCTS),m)
     M_OBJS += cpqfc.o
     endif
endif
```

3. Around line 703 (if using RedHat x6.2) or 760 (if using RedHat v7.0), search for the following code block:

```
megaraid.o: megaraid.c
    $(CC) $(CFLAGS) -c megaraid.c

scsi_mod.o: $(MIX_OBJS) hosts.o scsi.o scsi_ioctl.o constants.o \
        scsicam.o scsi_proc.o scsi_error.o scsi_obsolete.o
           scsi_queue.o
```

Insert the following lines between the `megaraid` and `scsi_mod` sections:

```
qla2x00.o: ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c
        $(CC) $(CFLAGS) -c qla2100.c -o qla2x00.o
```

The result should look like the following:

```
megaraid.o: megaraid.c
    $(CC) $(CFLAGS) -c megaraid.c
qla2x00.o: ql2100_fw.h ql2200_fw.h qla2100.h qla2100.c
        $(CC) $(CFLAGS) -c qla2100.c -o qla2x00.o
scsi_mod.o: $(MIX_OBJS) hosts.o scsi.o scsi_ioctl.o constants.o \
        scsicam.o scsi_proc.o scsi_error.o scsi_obsolete.o
           scsi_queue.o
```

Edit hosts.c

Depending on the Linux distribution and the kernel version, the lines indicated in this section may differ.

Edit **hosts.c** (located in **/usr/src/linux/drivers/scsi**) as follows:

1. **`vi hosts.c`**
2. Around line 190 (if using RedHat x6.2) or 198 (if using RedHat v7.0), search for the following code block:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
#include "qlogicfc.h"
#endif

#ifdef CONFIG_SCSI_SEAGATE
#include "seagate.h"
#endif
```

Insert the following lines between the `qlogicfc` and `seagate` sections:

```
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
#include "qla2100.h"
#endif
```

The result should look like the following:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
#include "qlogicfc.h"
#endif

#ifdef CONFIG_SCSI_QLOGIC_QLA2100
#include "qla2100.h"
#endif

#ifdef CONFIG_SCSI_SEAGATE
#include "seagate.h"
#endif
```

3. Around line 515 (if using RedHat v6.2) or 542 (if using RedHat v7.0), search for the following code block:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
 QLOGICFC,
#endif
#ifdef CONFIG_SCSI_PAS16
 MV_PAS16,
#endif
```

Insert the following lines in between the `endif` and `ifdef`:

```
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
 QLA2100_LINUX_TEMPLATE,
#endif
```

The result should look like the following:

```
#ifdef CONFIG_SCSI_QLOGIC_FC
 QLOGICFC,
#endif
#ifdef CONFIG_SCSI_QLOGIC_QLA2100
 QLA2100_LINUX_TEMPLATE,
#endif
#ifdef CONFIG_SCSI_PAS16
 MV_PAS16,
#endif
```

Modular Driver Specifics

1. In order to load the driver automatically at boot time, **/etc/conf.modules** must be edited.

   ```
   vi /etc/conf.modules
   ```

   Add the appropriate entry:

System:	Enter:
UP	**alias scsi_hostadapter qla2x00**
SMP	**alias scsi_hostadapter qla2x00smp**

2. In a fabric environment, persistent binding must be enabled in the driver. Without a persistent binding mechanism, the host will be unable maintain persistent logical routing of the communication from a storage device object across the fabric to a Symmetrix volume. If the physical configuration of the switch is changed (for example, cables are swapped or the host is rebooted), the logical route becomes inconsistent, which could cause possible data corruption.

   To enable persistent binding, **/etc/conf.modules** must be edited. The initiator and target information necessary for persistent binding are logged in **/var/log/messages** when the driver loads. *grep* the file to obtain the initiator and target information:

   ```
   grep scsi-qla /var/log/messages
   ```

   The output will be similar to the following:

```
Jan 9 11:24:48 denali kernel: scsi-qla0-adapter-node=0000000000000000;
Jan 9 11:24:48 denali kernel: scsi-qla0-adapter-port=200000e08b009209;
Jan 9 11:24:48 denali kernel: scsi-qla0-target-0=500604800000000c;
```

   The above information is appended to **/etc/conf.modules**. An example of the **/etc/conf.modules** configuration file for a uni-processor system would be:

   ```
   more /etc/conf.modules
   ```

```
alias scsi_hostadapter aic7xxx
alias scsi_hostadapter qla2x00

options qla2x00 ql2xopts=scsi-qla0-node=0000000000000000\;scsi-qla0-
adapter=200000e08b009209\;scsi-qla0-target-0=500604800000000c\;
```

When configuring and manually loading the **qla2x00(smp)** module with persistent binding information, you must use **modprobe** rather than **insmod**. **modprobe** is a wrapper that is an extension to **insmod**. This utility will load the driver and automatically read the **/etc/conf.modules** configuration file.

3. Proceed to *Required Kernel Modifications* on page 60 to modify and compile the Linux kernel.

Static Driver Specifics

1. In a fabric environment, persistent binding must be enabled in the driver. Without a persistent binding mechanism, the host will be unable maintain persistent logical routing of the communication from a storage device object across the fabric to a Symmetrix volume. If the physical configuration of the switch is changed (for example, cables are swapped or the host is rebooted), the logical route becomes inconsistent, which could cause possible data corruption.

   If the **qla2x00(smp)** driver is built statically into the kernel, the persistent binding information must be added to **/etc/lilo.conf** rather than **/etc/conf.modules**. To obtain the initiator and target information, *grep* the **/var/log/messages** file as indicated previously for the modular driver:

   ```
   grep scsi-qla /var/log/messages
   ```

   The output will be similar to the following:

```
Jan 9 11:24:48 denali kernel:scsi-qla0-adapter-node=0000000000000000;
Jan 9 11:24:48 denali kernel: scsi-qla0-adapter-node=200000e08b009209;
Jan 9 11:24:48 denali kernel:scsi-qla0-target-0=500604800000000c;
```

   Append the obtained information to **/etc/lilo.conf** using the **ql2xopts** parameter to add the persistent binding information to the file.

   For example, **/etc/lilo.conf** would contain an entry similar to the following:

```
append = "ql2xopts = scsi-qla0-adapter-node=0000000000000000;
   scsi-qla0-adapter-port=200000e08b009209;
   scsi-qla0-target-0=500604800000000c;"
```

   Ensure that **mkinitrd /boot/initrd-$1.img $1** and **lilo** are invoked when compiling the kernel in order for the driver to be loaded correctly as static.

2. Proceed to *Required Kernel Modifications* on page 60 to modify and compile the Linux kernel.

Required Kernel Modifications

Make the following changes to the kernel.

Edit hosts.h

In order to see more than four SCSI hard drives, edit **hosts.h** as follows:

```
cd /usr/src/linux/drivers/scsi
vi hosts.h
```

The text of **hosts.h** differs depending on the RedHat distribution version:

- v6.2:

```
#define SD_EXTRA_DEVS     40
#else
#define SD_EXTRA_DEVS     4   (Change to 128)
#endif
```

- v7.0:

```
#define SD_EXTRA_DEVS     40   (Change to 128)
#define ST_EXTRA_DEVS     3
#define SR_EXTRA_DEVS     8
```

Edit scsi.c

To allow the host to access and utilize more than eight SCSI LUNs, edit **scsi.c** as follows:

```
cd /usr/src/linux/drivers/scsi
vi scsi.c
```

```
#ifdef CONFIG_SCSI_MULTI_LUN
static int max_scsi_luns = 8;  (Change to 128)
#else
static int max_scsi_luns = 1;
#endif
```

This can also be accomplished by appending a line to **/etc/lilo.conf**:

```
vi /etc/lilo.conf
```

Add this line:

```
append="max_scsi_luns=128"
```

Edit sg.h

The value for the SCSI generic buffer must be increased. Change to the appropriate directory and increase the value for **SG_SCATTER_SZ** to the maximum of **131072**.

```
cd /usr/src/linux/include/scsi
vi sg.h
```

Around line 195, look for the following code block:

```
#define SG_SCATTER_SZ (8 * 4096)  /* PAGE_SIZE not available to user */
/* Largest size (in bytes) a single scatter-gather list element can have.
   The value must be a power of 2 and <= (PAGE_SIZE * 32) [131072 bytes on
   i386]. The minimum value is PAGE_SIZE. If scatter-gather not supported
   by adapter then this value is the largest data block that can be
   read/written by a single scsi command. The user can find the value of
   PAGE_SIZE by calling getpagesize() defined in unistd.h . */
```

Change the value for **SG_SCATTER_SZ (8* 4096)** to (**32 * 4096)**.

After the kernel compilation and system reboot, using the following command can validate this change:

```
cat /proc/sys/kernel/sg-big-buff
```

Proceed to *Configuring and Compiling the Kernel* on page 64 to compile the Linux kernel.

Additional Notes

Dynamic Device Reconfiguration

Linux currently lacks a command built into the kernel that allows for a dynamic SCSI channel reconfiguration like **drvconfig** or **ioscan**. Depending upon the hardware configuration and on whether the QLogic driver has been loaded into the kernel as a module, the QLogic driver can be unloaded and then reloaded. If the QLogic driver has been loaded as a module, this unloading and reloading will cause a scan of the SCSI bus. The driver should detect the newly added device(s). If the QLogic driver is loaded into the kernel statically, the system will need to be rebooted in order for the devices to be detected reliably.

The **modprobe** command is a wrapper or an extension to **insmod**. It uses and maintains a set of files that describe all the modules that are available for the current kernel in **/lib/modules**. modprobe can be used to load and unload an entire set of modules in addition to just a single module. Both the loading and unloading of the QLogic modular driver is done with **modprobe**.

The unloading of the module is done with **modprobe -r qla2x00(smp). modprobe** with the **-r** switch used to unload the loadable modules from the running kernel if they are not in use and if other modules are not dependent upon them.

In the standard Linux v2.2.x kernel, the SCSI addresses are not used in the device names as they are in other types of UNIX (Sun, SGI, HP-UX, and BSD, for example). Block device filenames take the form **/dev/sdln**, where **l** is the letter denoting the physical drive and **n** is the number denoting the partition on that physical drive. Device names and minor numbers are assigned dynamically at boot time or device loading time in the order of discovery.

Depending upon the hardware configuration, if a device is added and the system rebooted, the device numbering can and will change. This change makes the host's mount table inaccurate. For the most consistent results and to reduce the possibility of mount table inaccuracies, new devices should be appended to the list of already attached devices. For example, if the host contains multiple HBAs, it would be best to append the new device to the last HBA and to the end of the device list attached to that HBA. This would eliminate the need to alter the pre-existing entries in the mount table, since the new device could be appended to that as well. If a new device were added to the first out of two HBAs and the system rebooted, the devices would all shift by one number and the mount table entries would also need to be shifted by one device. If there is only one HBA, the new device can more easily be appended to the list of the regularly attached devices and the mount table altered accordingly.

tasks.h

There is a limit to the number of processes that can run simultaneously on a host. The default is set to 512. If the environment requires it, this value can be increased up to 4092.

To change the value:

1. **`cd /usr/src/linux/include/linux`**
2. **`vi tasks.h`**
3. Line 14 is:

```
#define NR_TASKS   512     /*On x86 Max 4092, or 4090 w/ APM configured.*/
```

Change the value of `512` to whatever is required up to a maximum of 4092; then proceed to *Configuring and Compiling the Kernel* on page 64.

BLIST_NOLUN

Hosts with a GEM SCSI chip on the motherboard may incorrectly report multiple processors when booting. This incorrect reporting can be fixed by adding an entry to **BLIST_NOLUN** in the file **scsi.c**.

For example, in the NEC ExpressServer 5800 120Ra2, the GEM chip incorrectly reports back 8 LUNs instead of 1 LUN as it should. This problem can be corrected by editing **scsi.c** as follows:

1. **`cd /usr/src/linux/drivers/scsi`**
2. **`vi scsi.c`**
3. Search for the following:

```
   static struct dev_info device_list [] =
{"Aashima", "IMAGERY 2400sp", "1.03", BLIST_NOLUN}, /*Locks up if
   polled for lun != 0 */
{"CHINON", "CD-ROM CDS-431", "H42", BLIST_NOLUN}, /*Locks up if polled
   for lun != 0*/
```

where:

- `Aashima` = Make
- `IMAGERY2 2400sp` = Model
- `1.03` = firmware
- `BLIST_NOLUN` = flag

4. Go to the end of the `BLIST_NOLUN` list and add the device. For example, in the case of the NEC ExpressServer 5800 120Ra2, the entry would be similar to the following:

```
{"NEC", "GEM312R-G7CGQ", "4.1a", BLIST_NOLUN}, /*Reports back
   incorrect LUNs*/
```

5. After this addition is made; then proceed to *Configuring and Compiling the Kernel* on page 64.

Configuring and Compiling the Kernel

If this is a new installation, proceed to New Installation later on this page. If this is an upgrade to a newer kernel revision, go to *Kernel Upgrade* on page 67.

New Installation

Follow these steps to configure the host in a new installation:

1. Prior to configuring the kernel, change to the appropriate directory:

   ```
   cd /usr/src/linux
   ```

2. Configure the kernel:

   ```
   make config/menuconfig/xconfig
   ```

3. From the main menu, select the **Processor type and features** menu.

   a. Select the appropriate **Processor family** for the processors in the system:

   - 386
   - 486/Cx486
   - 586/K5/5x86/6x86
   - Pentium/K6/TSC
   - PPro/6x86MX/PII
   - PIII/Xeon/Deschutes

   b. Enable **Symmetric multi-processing support** as static ***only*** if the system being used has multiple processors.

   c. Select **<Exit>** to return to the Main Menu.

4. From the main menu, select the **SCSI support** menu (Figure 3 on page -65).

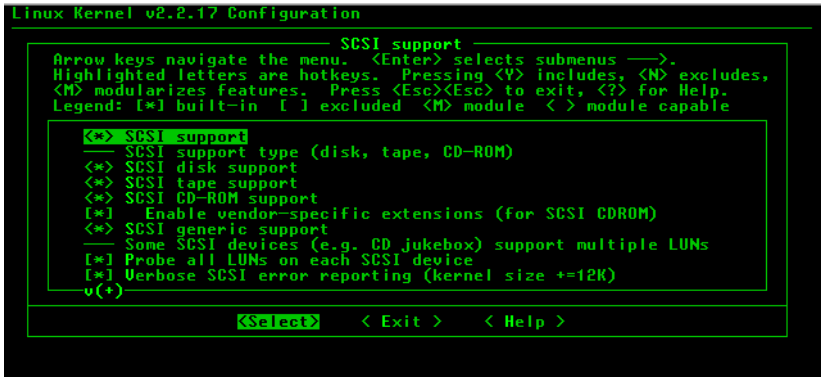
Linux Kernel v2.2.17 Configuration
SCSI support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> SCSI support
--- SCSI support type (disk, tape, CD-ROM)
<*> SCSI disk support
<*> SCSI tape support
<*> SCSI CD-ROM support
[*] Enable vendor-specific extensions (for SCSI CDROM)
<*> SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*] Probe all LUNs on each SCSI device
[*] Verbose SCSI error reporting (kernel size +=12K)
v(+)
<Select> < Exit > < Help >

Figure 3 **SCSI Support Menu**

Select **SCSI support** as static (* or **y**).

Under

---SCSI support type (disk, tape, CD-ROM)

select the options **SCSI disk support** and **SCSI generic support** as static (* or **y**).

Under

---Some SCSI devices (e.g. CD jukebox) support multiple LUNs

select **Probe all LUNs on each SCSI device** as static (* or **y**).

The **qlogicfc** driver must be disabled in the kernel to eliminate any possible conflicts between the **qlogicfc** and **qla2x00(smp)** drivers.

Under

SCSI low-level drivers

deselect **QLogic ISP FC SCSI support**.

deselect **Qlogic QLA 2x00 FC SCSI support**.

If Method Two was used to incorporate the QLogic driver into the kernel (refer to *Installation Methods* on page 45), you must select the QLogic driver in **menuconfig** as static or modular.

Under

SCSI low-level drivers

select **EMC-approved QLogic QLA2x00F Driver** as modular (**M** or **m**) or static (***** or **y**), as required. (Refer to the above note.)

Select **Exit** to return to the Main Menu.

From the Main Menu, select **Exit**; then select **Yes** to save your configuration in response to the question `Do you wish to save your new kernel configuration?`

The configuration will be saved to the file **/usr/src/linux/.config**.

5. Compile the kernel:

```
make dep
make clean
make install
make modules
make modules_install
cd /boot
```

6. Build a ramdisk image:

```
mkinitrd /boot/initrd-$1.img $1
```

(where **$1** is the kernel version)

For example:

```
mkinitrd /boot/initrd-2.2.17.img 2.2.17
```

The **mkinitrd** command is used to create a new ramdisk image and is dependent upon the loopback device being available. If the loopback driver is selected in the kernel as a module, ensure that it is loaded with **#lsmod**. If the loopback driver was not loaded, insert it with **#insmod loop**.

---

7. Run **lilo** to update the loading map and reboot:

```
lilo -v
reboot
```

Kernel Upgrade

For a kernel upgrade, install Linux from the RedHat v6.2 CD, following the procedure in the installation guide. Partition the boot drive and select the packages and services necessary for the host. The host can be upgraded from RedHat v6.2 to up to v2.2.17 of the Linux kernel.

To upgrade the host to a newer version of the kernel, the kernel must be downloaded, from: **http://www.kernel.org/mirrors/**.

1. Prepare the host for the new version of the kernel:

```
cd /usr/src
rm -f linux
```

2. Untar the kernel and create a directory for it:

```
tar zxvf linux-$1.tar.gz
mv linux linux-$1
ln -s linux-$1 linux
cd linux
```

(where `$1` is the kernel version)

3. Delete stale **.o** files and dependencies:

```
make mrproper
```

4. Configure the kernel:

```
make config/menuconfig/xconfig
```

From the main menu, select the **SCSI support** menu (Figure 4):

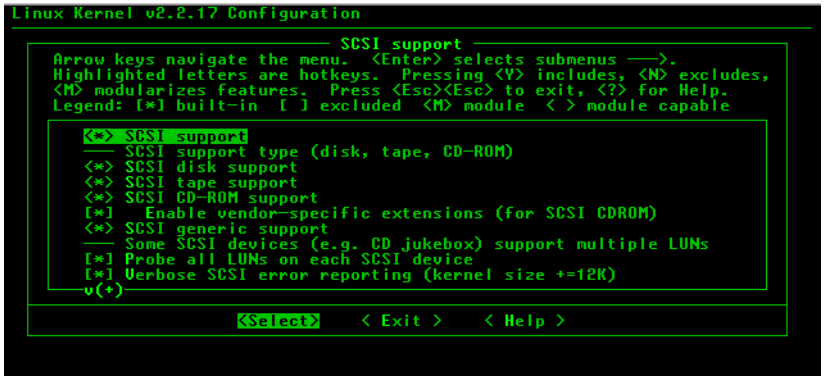
Linux Kernel v2.2.17 Configuration
SCSI support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> SCSI support
--- SCSI support type (disk, tape, CD-ROM)
<*> SCSI disk support
<*> SCSI tape support
<*> SCSI CD-ROM support
[*] Enable vendor-specific extensions (for SCSI CDROM)
<*> SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*] Probe all LUNs on each SCSI device
[*] Verbose SCSI error reporting (kernel size +=12K)
v(+)
<Select> < Exit > < Help >

Figure 4 **SCSI Support Menu**

Select **SCSI support** as static (* or **y**).

Under

---SCSI support type (disk, tape, CD-ROM)

select the options **SCSI disk support** and **SCSI generic support** as static (***** or **y**).

Under

---Some SCSI devices (e.g. CD jukebox) support multiple LUNs

select **Probe all LUNs on each SCSI device** as static (***** or **y**).

The **qlogicfc** driver must be disabled in the kernel to eliminate any possible conflicts between the **qlogicfc** and **qla2x00(smp)** drivers.

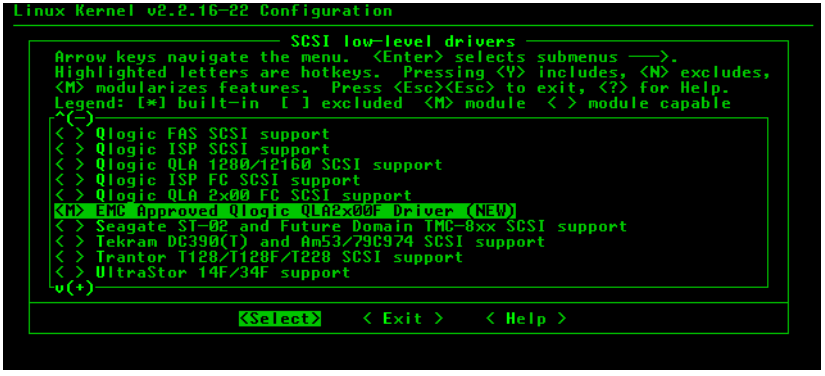
Linux Kernel v2.2.16-22 Configuration
SCSI low-level drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
< > Qlogic FAS SCSI support
< > Qlogic ISP SCSI support
< > Qlogic QLA 1280/12160 SCSI support
< > Qlogic ISP FC SCSI support
< > Qlogic QLA 2x00 FC SCSI support
<M> EMC Approved Qlogic QLA2x00F Driver (NEW)
< > Seagate ST-02 and Future Domain TMC-8xx SCSI support
< > Tekram DC390(T) and Am53/79C974 SCSI support
< > Trantor T128/T128F/T228 SCSI support
< > UltraStor 14F/34F support
<Select> < Exit > < Help >

Figure 5 **SCSI Low-Level Drivers Menu**

Under

SCSI low-level drivers

deselect **QLogic ISP FC SCSI support**.

deselect **Qlogic QLA 2x00 FC SCSI support**.

If Method Two was used to incorporate the QLogic driver into the kernel (refer to *Installation Methods* on page 45), you must select the QLogic driver in **menuconfig** as static or modular.

Under

SCSI low-level drivers

select **EMC-approved QLogic QLA2x00F Driver** as modular (**M** or **m**) or static (***** or **y**), as required. (Refer to the above note.)

5. Compile the kernel:

```
make dep
make clean
make install
make modules
make modules_install
cd /boot
```

6. Add the new kernel version to **lilo.conf** so it will be bootable:

```
vi /etc/lilo.conf
```

For example:

```
image=/boot/vmlinuz-2.2.17
label=linux-2.2.17
root=/dev/sda1
initrd=/boot/initrd-2.2.17.img
read-only
```

7. Build a ramdisk image:

```
mkinitrd /boot/initrd-$1.img $1
```

(where `$1` is the kernel version)

For example:

```
mkinitrd /boot/initrd-2.2.17.img 2.2.17
```

The **mkinitrd** command is used to create a new ramdisk image and is dependent upon the loopback device being available. If the loopback driver is selected in the kernel as a module, ensure that it is loaded with **#lsmod**. If the loopback driver was not loaded, insert it with **#insmod loop**.

---

8. Run **lilo** to update the loading map and reboot:

```
lilo -v
reboot
```

Incorporating Symmetrix Fibre Channel

Once the Symmetrix has devices assigned with device addresses and the directors have been switched on line, the host can see all the target Symmetrix devices assigned to that host interface. Devices are presented to the host in the same manner as devices accessed through a standard SCSI interface.

Partitioning and Labeling a New Device

Block device filenames take the form **/dev/sd*ln***, where ***l*** is the letter denoting the physical drive and ***n*** is the number denoting the partition on that physical drive. Usually, the partition number is not included when the device corresponds to the entire drive.

Following this format, the filenames would appear as follows:

/dev/sd[a-d][a-z][1-15]

Linux supports a total of 128 SCSI devices per system. As a result, the host could support disk devices from **/dev/sda** through **/dev/sddx**.

Linux supports one to 15 partitions per disk device. Partitions 1 through 4 are the primary partitions, while partitions 5 and greater are the logical or extended partitions. (These limitations are specific to the Intel platform.) By default, slices are not used in Linux.

True raw devices have not yet been incorporated into the kernel.

The corresponding character device filenames take the form **/dev/sg[a-d][a-z]**, where **sg** is one of four high-level device drivers in the SCSI subsystem.

The four high-level device drivers are:

sg (SCSI generic interface)

sd (direct access; disks, for example)

st (tapes)

sr (data CD-ROMs)

The **sg** driver is a character-based device, while the other three drivers are block-based devices. **sg** is used primarily for scanners, CD writers, printers, and raw access, although **sg** is not a true raw device. The **sg** device files are dynamically mapped to SCSI IDs/LUNs on the SCSI bus, starting with the first SCSI controller.

The Linux kernel assigns minor SCSI device numbers dynamically by assigning them only to devices that are actually connected to the host in the order of their SCSI IDs. This means that connecting an external SCSI device can change the minor numbers of all the internal SCSI devices with a higher SCSI ID.

Partitioning Symmetrix Devices

Once the QLogic driver is loaded, the Symmetrix devices will be represented in **/proc/scsi/scsi**. They can be viewed by using **more /proc/scsi/scsi**. Only **/dev/sda** through **/dev/sdp** and **/dev/sga** through **/dev/sgh** will be created by default upon kernel compilation. In order to create subsequent device files for the Symmetrix devices attached to the host, the root user will need to create them via the **mknod** command.

Using the **fdisk** command allows multiple partitions to reside on a single disk device.

To partition one or more Symmetrix devices for the use of Linux, log in as *root* and follow the procedure below:

1. Type **`fdisk /dev/sdb`** ENTER.
2. If you want to display a list of options, type **`m`** ENTER (for help) to display the following:

```
Command action
 a     toggle a bootable flag
 b     edit bsd disklabel
 c     toggle the DOS compatibility flag
 d     delete a partition
 l     list known partition types
 m     print this menu
 n     add a new partition
 o     create a new empty DOS partition table
 p     print the partition table
 q     quit without saving changes
 s     create a new empty Sun disklabel
 t     change a partition's system ID
 u     change display/entry units
 v     verify the partition table
 w     write table to disk and exit
 x     extra functionality (experts only)
```

To display a list of advanced options, type **x** ENTER (for expert functionality), then **m** ENTER to display the following:

```
Command action
 b     move beginning of data in a partition
 c     change number of cylinders
 d     print the raw data in the partition table
 e     list extended partitions
 g     create an IRIX partition table
 h     change number of heads
 m     print this menu
 p     print the partition table
 q     quit without saving changes
 r     return to main menu
 s     change number of sectors
 v     verify the partition table
 w     write table to disk and exit
```

If you display the above advanced menu, type **m** ENTER when finished viewing the menu, to return to the main menu.

3. On the main menu, select the options you need to create and label a new partition:
   - Type **n** ENTER to add a new partition.
   - Type **t** ENTER to change a partition's system ID.
   - You can type **L** ENTER for a list of hex codes for system IDs. (Linux is type 83. Linux swap is type 82.)
   - Type **p** ENTER to print the partition table and to verify that the selections look correct.
   - Type **w** ENTER to write the table to disk and exit **fdisk**.
4. Repeat step 3 for each new Symmetrix device.

cfdisk is a curses-based disk partition table manipulator for Linux that is used instead of **fdisk**.) If used with the **-z** option, **cfdisk** allows the device's partition table to be zeroed out. The use of **cfdisk -z** is required if you are converting a device previously used by another host.

Example: **`cfdisk -z /dev/sdb`**

The resulting display would be similar to the following:

```
                    cfdisk 0.8n

                 Disk Drive: /dev/sdb
       Heads: 255   Sectors per Track: 63   Cylinders: 550

  Name      Flags     Part Type     FS Type     [Label]     Size (MB)
------------------------------------------------------------------------

               Pri/Log  Free Space              4314.34




  [  Help  ]  [  New   ]  [ Print  ]  [  Quit  ]  [ Units  ]
  [ Write  ]
```

CAUTION

Changing the partition table results in loss of any previous data.

Select **[New]**. The following options appear at the bottom of the window:

[Primary] [Logical] [Cancel]

Select **[Primary]** to create a primary Linux partition and select the size of the partition that is required.

In this case, to use the entire device, select:

Size (in MB): 4314.34

Once the above step is completed, the options listed at the bottom will change to the following:

```
                        cfdisk 0.8n

                     Disk Drive: /dev/sdb
          Heads: 255   Sectors per Track: 63   Cylinders: 550

  Name      Flags     Part Type     FS Type     [Label]     Size (MB)
------------------------------------------------------------------------

  sdb1                Primary      Linux                   4314.34




  [  Bootable  ]  [  Delete   ]  [ Help  ]  [  Maximize  ]  [ Print  ]
  [ Quit  ] [  Type  ][  Units  ] [  Write  ]
```

cfdisk will automatically select Linux as the file system type. If a different type of file system or a swap partition is required, select **[Type]** and choose the necessary file system type.

Select **[Write]** to write the partition table to the disk. **cfdisk** will prompt the user with:

```
"Are you sure you want to write the partition table to disk (yes or no)?
   Warning!! This may destroy data on your disk!"
```

Type **yes** ENTER to write the partition table. Once the partition table has been written to the disk, select **[Quit]** to exit **cfdisk**.

Inaccessible HBA Fix

In the event that the HBA becomes inaccessible, follow the steps listed below:

1. Unplug the fiber cables from the host.
2. Reboot the host.
3. Upon rebooting, press ALT-Q when prompted to enter the **Fast!Util** menu.
4. From the **Fast!Util Options** menu, select **Configuration Settings**.
5. On the **Configuration Settings** menu, select **Restore Default Settings**.
6. Return to the procedure under *Setting the NVRAM Variables* on page 42 to reestablish the EMC-recommended adapter settings.

7. Save the settings for the HBA installed in the host and exit **Fast!Util**.
8. Plug the fiber cables back into the HBA(s) in the host and reboot.
9. Upon rebooting, press ALT-Q when prompted to enter the **Fast!Util** menu.
10. From the **Fast!Util Options** menu, select **Fibre Disk Utility**.
11. Follow the instructions from *Verifying the Disk Media* on page 44 to ensure that the HBA has been reset and can now access all of the Symmetrix devices.
12. Exit out of the Fibre Disk utility and reboot the host.
13. Once the Symmetrix devices are partitioned and labeled appropriately, proceed with *Creating and Mounting a File System* on page 75.

Creating and Mounting a File System

To create standard UNIX file systems under Linux, log in as *root* and proceed as follows for each previously labeled device.

Create a New File System

Once each Symmetrix disk device has been formatted, partitioned, and labeled, a file system needs to be created in order to make the partition usable. Use the **mkfs** command to create a file system.

For a single partition, use **`mkfs /dev/sdb1`**.

You will be prompted to confirm the construction of a new file system. Type **`Y`** ENTER.

Create a Mount Directory

Once the file system for each Symmetrix device has been created, a mount directory for each device must be created.

To create a mount directory, use **`mkdir /mnt/sdb`**, where **`/mnt/sdb`** is the complete path for the new file system.

By default, **/mnt** is created upon installation, but a separate directory can be created in which to mount the devices' file systems.

Mount the File System

To mount each file system, enter a statement similar to the following:

```
mount -t ext2 /dev/sdb1 /mnt/sdb
```

This will mount the raw device **/dev/sdb1** onto the mount point **/mnt/sdb**.

System and Error Messages

Linux logs system and error messages to a file called **/var/log/messages**. This file includes SCSI error messages. Linux also logs boot messages in a file called **/var/log/boot.log**. The host will also display these messages at the system console.